Properties of a lambda calculus with definitions

Herman Geuvers
Radboud University Nijmegen
and
Eindhoven University of Technology
The Netherlands

1 Introduction

In this short note we prove the meta-theoretic properties of the typed λ-calculus λD, that were announced and used in the book [7]. Many of them are a straightforward extension of the corresponding properties of λC or for Pure Type Systems (see [1, 5, 3]) Since we now have *definitions*, there are more properties to be stated and proved and the inductive proofs of the known statements have additional cases to consider. In particular, we now have statements about definition unfolding and δ-conversion. Below we give the rules of λD, without explanation or motivation (for which we refer to [7]), and then we state and prove the relevant lemmas for λD.

The statements and proofs for definitions are very much like the ones in [8]. As λD is not exactly the same as the system in that paper, we give full proofs of the properties below. Concerning δ-conversion, we only give the relevant normalization and confluence properties.

2 Definitions

Definition 2.1. (Expressions of $\lambda\mathrm{D}_0$ and λD, $\mathcal{E}_{\lambda D}$)
The set $\mathcal{E}_{\lambda D}$ of *expressions* of $\lambda\mathrm{D}_0$ (and λD), is defined by:

$$\mathcal{E}_{\lambda D} = V \mid \Box \mid * \mid (\mathcal{E}_{\lambda D}\, \mathcal{E}_{\lambda D}) \mid (\lambda V : \mathcal{E}_{\lambda D} \,.\, \mathcal{E}_{\lambda D}) \mid (\Pi V : \mathcal{E}_{\lambda D} \,.\, \mathcal{E}_{\lambda D}) \mid C(\overline{\mathcal{E}_{\lambda D}})$$

The set of expressions of λD is the same as the set of expressions of $\lambda\mathrm{D}_0$: $\mathcal{E}_{\lambda D}$. So it is important to note that we do not consider '$\perp\!\!\!\perp$' as an expression: it is a meta-symbol in judgments, on a par with e.g. ':=' and '$\rhd$'.

Definition 2.2. (Descriptive definitions in $\lambda\mathrm{D}_0$; environment)
(1) A *(descriptive) definition* in $\lambda\mathrm{D}_0$ has the form
$\overline{x} : \overline{A} \rhd a(\overline{x}) := M : N$,
with all $x_i \in V$, $a \in C$, and all $A_i, M, N \in \mathcal{E}_{\lambda D}$.
(2) An *environment* Δ is a finite (empty or non-empty) list of definitions.

Definition 2.3. (Elements of a definition)
Let $\mathcal{D} \equiv \overline{x} : \overline{A} \rhd a(\overline{x}) := M : N$ be a definition. Then:

- $\overline{x} : \overline{A}$ is the *context* of $\mathcal{D}$.
- a is the *defined constant* of $\mathcal{D}$, with $\overline{x}$ as parameter list.
- $a(\overline{x})$ is the *definiendum* of $\mathcal{D}$.
- $M : N$ is the *statement* of $\mathcal{D}$, M is the *definiens* or the *body* of $\mathcal{D}$, and N is the *type* of $\mathcal{D}$.

Recall that the prominent expressive entity in λC is the *judgment* (also called *typing judgment*), having the form

$\Gamma \vdash M : N$.

In the presence of definitions, such a judgment may well depend on one or more defined constants. In particular, M and N, but also the types in the context Γ may contain one or

more constants: a_1, a_2, ... So each judgment must have the possibility to be *preceded by* an environment, which is a list $\Delta \equiv \mathcal{D}_1, \ldots, \mathcal{D}_k$ of definitions.

Let's use meta-symbol ';' for the separation between an environment and a judgment. Then we obtain a new general format for *a judgment with definitions*:

Definition 2.4. (Judgment with definitions; extended judgment)
A *judgment with definitions* or *extended judgment* has the form

$\Delta \ ; \ \Gamma \ \vdash \ M : N,$

with Δ an environment, Γ a context and $M, N \in \mathcal{E}_{\lambda D}$.

We pronounce this as: 'M has type N in environment Δ and context Γ.'

By abuse of language, we will still use the simple word 'judgment' for such a 'judgment with definitions'; sometimes we'll speak of *extended judgments* to distinguish them from the judgments without definitions, as presented in the previous chapters.

By writing out the environment Δ and the context Γ we obtain:

$\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k \ ; \ x_1 : A_1, \ldots, x_n : A_n \ \vdash \ M : N.$

So in this format, the basic statement $M : N$ has been 'decorated' at the front with a list Δ and a list Γ:

(1) the environment Δ binding the *constants* occurring in $M : N$,
(2) the context Γ binding the *free variables* occurring in $M : N$.

Definition 2.5. (One-step definition unfolding; one-step δ-reduction, $\overset{\Delta}{\to}$)
If $\Gamma \ \rhd \ a(\overline{x}) \ := \ M : N$ is an element of environment Δ, then
(1) (Basis) $a(\overline{U}) \overset{\Delta}{\to} M[\overline{x} := \overline{U}]$,
(2) (Compatibility) If $M \overset{\Delta}{\to} M'$, then $ML \overset{\Delta}{\to} M'L$, $LM \overset{\Delta}{\to} LM'$, $\lambda x\,.\,M \overset{\Delta}{\to} \lambda x\,.\,M'$ and $b(\ldots, M, \ldots) \overset{\Delta}{\to} b(\ldots, M', \ldots)$.

Definition 2.6. (δ-reduction (zero-or-more-step), $\overset{\Delta}{\twoheadrightarrow}$)
$M \overset{\Delta}{\twoheadrightarrow} N$ if there is an n and there are expressions M_0 to M_n such that $M_0 \equiv M$, $M_n \equiv N$ and for all i such that $0 \leq i < n$:

$$M_i \overset{\Delta}{\to} M_{i+1} \ .$$

Definition 2.7. (δ-conversion, $\overset{\Delta}{=}$)
$M \overset{\Delta}{=} N$ (to be read as: 'M and N are convertible with respect to Δ') if there is an n and there are expressions M_0 to M_n such that $M_0 \equiv M$, $M_n \equiv N$ and for all i such that $0 \leq i < n$:

$$\textit{either } M_i \overset{\Delta}{\to} M_{i+1} \textit{ or } M_{i+1} \overset{\Delta}{\to} M_i \ .$$

So M and N are δ-convertible (or $M \overset{\Delta}{=} N$) if the one can be obtained from the other by successively folding or unfolding a number of definitions occurring in it.

Remark 2.8. *The relation* $\overset{\Delta}{=}$ *is an* equivalence relation *on expressions, just as* $=_\beta$*. It is*

- reflexive*: for all* L*:* $L \ \overset{\Delta}{=} \ L$*,*
- symmetric*: for all* L*,* M*: if* $L \ \overset{\Delta}{=} \ M$*, then* $M \ \overset{\Delta}{=} \ L$*, and*

– transitive*: for all L, M and N: if $L \stackrel{\Delta}{=} M$ and $M \stackrel{\Delta}{=} N$, then $L \stackrel{\Delta}{=} N$.*

Comparable to what is standard for β-reduction, we define the *δ-normal form* of an expression, with respect to an environment Δ.

Definition 2.9. (Unfoldable, δ-normal form, δ-nf)
Let Δ be an environment.

(1) A constant a is *unfoldable* with respect to Δ, if a is bound to a descriptive definition in Δ, say: $\overline{x} : \overline{A} \rhd a(\overline{x}) := M : N$.

(2) *K is in δ-normal form* (or: is in δ-nf) with respect to Δ, if there occurs no constant in K that is unfoldable with respect to Δ.

(3) *K has a δ-normal form* (has a δ-nf) with respect to Δ, if there is an L in δ-nf with respect to Δ such that $K \stackrel{\Delta}{=} L$. One also says in this case: K is δ-normalizing, and L is a δ-normal form of K (with respect to Δ).

For completeness, we first give the derivation rules for λC:

$$(sort) \quad \emptyset \vdash * : \Box$$

$$(var) \ \frac{\Gamma \vdash A : s}{\Gamma,\ x : A \vdash x : A} \quad \text{if } x \notin \Gamma$$

$$(weak) \ \frac{\Gamma \vdash A : B \quad \Gamma \vdash C : s}{\Gamma,\ x : C \vdash A : B} \quad \text{if } x \notin \Gamma$$

$$(form) \ \frac{\Gamma \vdash A : s_1 \quad \Gamma,\ x : A \vdash B : s_2}{\Gamma \vdash \Pi x : A\,.\,B \ : \ s_2}$$

$$(appl) \ \frac{\Gamma \vdash M \ : \ \Pi x : A\,.\,B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN \ : \ B[x := N]}$$

$$(abst) \ \frac{\Gamma,\ x : A \vdash M : B \qquad \Gamma \vdash \Pi x : A\,.\,B \ : \ s}{\Gamma \vdash \lambda x : A\,.\,M \ : \ \Pi x : A\,.\,B}$$

$$(conv) \ \frac{\Gamma \vdash A : B \quad \Gamma \vdash B' : s}{\Gamma \vdash A : B'} \quad \text{if } B =_\beta B'$$

Definition 2.10. The *derivation rules for λD* are as follows. All the rules that do not involve $\perp\!\!\!\perp$ are *derivation rules for λD_0*.

$$(sort)\quad \emptyset\ ;\ \emptyset\ \vdash\ *:\Box$$

$$(var)\ \frac{\Delta\ ;\ \Gamma\ \vdash\ A:s}{\Delta\ ;\ \Gamma,\ x:A\ \vdash\ x:A}\ \text{if } x\notin\Gamma$$

$$(weak)\ \frac{\Delta\ ;\ \Gamma\ \vdash\ A:B\quad \Delta\ ;\ \Gamma\ \vdash\ C:s}{\Delta\ ;\ \Gamma,\ x:C\ \vdash\ A:B}\ \text{if } x\notin\Gamma$$

$$(form)\ \frac{\Delta\ ;\ \Gamma\ \vdash\ A:s_1\quad \Delta\ ;\ \Gamma,\ x:A\ \vdash\ B:s_2}{\Delta\ ;\ \Gamma\ \vdash\ \Pi x:A\,.\,B\ :\ s_2}$$

$$(appl)\ \frac{\Delta\ ;\ \Gamma\ \vdash\ M\ :\ \Pi x:A\,.\,B\qquad \Delta\ ;\ \Gamma\ \vdash\ N:A}{\Delta\ ;\ \Gamma\ \vdash\ MN\ :\ B[x:=N]}$$

$$(abst)\ \frac{\Delta\ ;\ \Gamma,\ x:A\ \vdash\ M:B\qquad \Delta\ ;\ \Gamma\ \vdash\ \Pi x:A\,.\,B\ :\ s}{\Delta\ ;\ \Gamma\ \vdash\ \lambda x:A\,.\,M\ :\ \Pi x:A\,.\,B}$$

$$(conv)\ \frac{\Delta\ ;\ \Gamma\ \vdash\ A:B\quad \Delta\ ;\ \Gamma\ \vdash\ B':s}{\Delta\ ;\ \Gamma\ \vdash\ A:B'}\ \text{if } B\stackrel{\Delta}{=}_\beta B'$$

$$(def)\ \frac{\Delta\ ;\ \Gamma\ \vdash\ K:L\qquad \Delta\ ;\ \overline{x}:\overline{A}\ \vdash\ M:N}{\Delta\ ,\ \overline{x}:\overline{A}\ \rhd\ a(\overline{x}):=M:N\ ;\ \Gamma\ \vdash\ K:L}\ \text{if } a\notin\Delta$$

$$(def\text{-}prim)\ \frac{\Delta\ ;\ \Gamma\ \vdash\ K:L\qquad \Delta\ ;\ \overline{x}:\overline{A}\ \vdash\ N:s}{\Delta\ ,\ \overline{x}:\overline{A}\ \rhd\ a(\overline{x}):=\perp\!\!\!\perp:N\ ;\ \Gamma\vdash K:L}\ \text{if } a\notin\Delta$$

$$(inst)\ \frac{\Delta;\Gamma\vdash *:\Box\quad \Delta;\Gamma\vdash \overline{U}:\overline{A[\overline{x}:=\overline{U}]}}{\Delta\ ;\ \Gamma\ \vdash\ a(\overline{U}):N[\overline{x}:=\overline{U}]}\ \text{if } \overline{x}:\overline{A}\rhd a(\overline{x}):=M:N\in\Delta$$

$$(inst\text{-}prim)\ \frac{\Delta;\Gamma\vdash *:\Box\quad \Delta;\Gamma\vdash \overline{U}:\overline{A[\overline{x}:=\overline{U}]}}{\Delta\ ;\ \Gamma\ \vdash\ a(\overline{U}):N[\overline{x}:=\overline{U}]}\ \text{if } \overline{x}:\overline{A}\rhd a(\overline{x}):=\perp\!\!\!\perp:N\in\Delta$$

3 Basic meta-theoretic properties

First, we establish that λD_0 is indeed an extension of λC and that λD, in its turn, extends λD_0. Each of these three systems has its own notion of derivability, implied by the specific rules of each system. By abuse of notation, we use the symbol '$\vdash$' for each of these three notions. If this may cause confusion, we add a phrase to mend this, such as: '$\Gamma;\Delta\vdash K:L$ in λD'.

Almost all the properties we prove are by induction on the derivation, doing a case distinction according to the last rule applied. For λD, this amounts to 11 cases. Often, these cases are straightforward: apply the induction hypothesis to the premises of the rule, then apply the rule, and we obtain exactly the conclusion that we need. These type of cases will not be detailed, but we will just say that these case are *immediate by induction on the derivation.* So we only treat the cases where some additional steps are needed, sometimes using an auxiliary property that we will then refer to explicitly. We will abbreviate *induction hypothesis* to *IH.*

Lemma 3.1. *(Inclusion of λC in λD_0, and of λD_0 in λD.)*

1. *If $\Gamma \vdash K : L$ in λC, then $\emptyset; \Gamma \vdash K : L$ in λD_0; and*
2. *If $\Delta; \Gamma \vdash K : L$ in λD_0, then $\Delta; \Gamma \vdash K : L$ in λD.*

Proof. Each of the cases in the two proofs is immediate by induction on the derivation. □

Next, we focus on λD; we use the notation $a(\overline{x}) := K/\!\!\perp$ to indicate that it does not matter whether a has been attached to a *description*, viz. K, or has *primitively* been defined as a constant.

The following definition is an extension of Definition 2.3.

Definition 3.2. (Elements of a definition in λD)
Let $\mathcal{D} \equiv \Gamma \rhd a(\overline{x}) := K/\!\!\perp : L$ be a definition.

- Γ is the *context* of $\mathcal{D}$.
- a is the *defined constant* of $\mathcal{D}$, with $\overline{x}$ as parameter list.
- $a(\overline{x})$ is the *definiendum* of $\mathcal{D}$.
- $K/\!\!\perp : L$ is the *statement* of $\mathcal{D}$.
- K resp. $\perp\!\!\!\perp$ is the *definiens* (also called: *body*) of $\mathcal{D}$.
- L is the *type* of $\mathcal{D}$.

If $\mathcal{D} \equiv \Gamma \rhd a(\overline{x}) := K : L$, then the definition is called a *descriptive* or *proper* definition, and a is a *proper* constant.
If $\mathcal{D} \equiv \Gamma \rhd a(\overline{x}) := \perp\!\!\!\perp : L$, then the definition is called a *primitive* definition, and a is a *primitive* constant.

The following lemma concerns (free) variable and constant occurrences:

Lemma 3.3. *(Free Variables and Constants Lemma)*
Suppose that

$$\Delta_1, \overline{x} : \overline{A} \rhd a(\overline{x}) := K/\!\!\perp : L, \Delta_2 \,; \Gamma \vdash M : N,$$

where $\Gamma \equiv \overline{y} : \overline{B}$ and $\Delta \equiv \Delta_1, \mathcal{D}, \Delta_2$. Then:

1. *For all i, $FV(A_i) \subseteq \{x_1, \ldots, x_{i-1}\}$; $FV(K), FV(L) \subseteq \{\overline{x}\}$.*
2. *For all j, $FV(B_j) \subseteq \{y_1, \ldots, y_{j-1}\}$; $FV(M), FV(N) \subseteq \{\overline{y}\}$.*
3. *Constant a does not occur in Δ_1.*
4. *If constant b occurs in $\overline{A}$, K or L, then $b \neq a$ and b is the defined constant of some $\mathcal{D} \in \Delta_1$.*
5. *If constant b occurs in $\overline{B}$, M or N, then b is the defined constant of some $\mathcal{D} \in \Delta$.*

Proof. The first two are by simultaneous induction on the derivation. All cases of (1) are immediate by induction on the derivation, except for (*def*) and (*def-prim*) where the induction hypothesis for (2) is used. The cases for (2) are the same as for the usual proof of the Free variables lemma in [1, 5, 3], except for the new cases (*def*), (*def-prim*), (*inst*) and (*inst-prim*). The cases (*def*) and (*def-prim*) are immediate by induction on the derivation. For the case (*inst*), if $M : N$ is $a'(\overline{U}) : N'[\overline{z} := \overline{U}]$, with $\overline{z} : \overline{C} \rhd a'(\overline{z}) := M' : N' \in \Delta$, then $FV(N') \subseteq \{\overline{z}\}$. So, $FV(N'[\overline{z} := \overline{U}]) \subseteq FV(\overline{U})$. By induction hypothesis, we know $FV(\overline{U}) \subseteq \{\overline{y}\}$, so we are done. The case (*inst-prim*) is the same as (*inst*).

Statements (3), (4), (5) are immediate by induction on the derivation. □

Lemma 3.4. *(Environment and Context Lemma)*

1. *If D is a derivation of $\Delta_1, \overline{x} : \overline{A} \rhd a(\overline{x}) := K : L, \Delta_2 \,;\, \Gamma \vdash M : N$, then there is a sub-derivation of D with conclusion $\Delta_1; \overline{x} : \overline{A} \vdash K : L$,*

2. *If D is a derivation of $\Delta_1, \overline{x} : \overline{A} \rhd a(\overline{x}) := \perp\!\!\!\perp : L, \Delta_2 \,;\, \Gamma \vdash M : N$, then there is a sub-derivation of D with conclusion $\Delta_1; \overline{x} : \overline{A} \vdash L : s$, for some sort s.*

3. *If D is a derivation of $\Delta \,;\, \Gamma_1, x : A, \Gamma_2 \vdash M : N$, then there is a sub-derivation of D with conclusion $\Delta'; \Gamma_1 \vdash A : s$, for some sort s and some Δ' which is a prefix of Δ.*

Proof. All three cases are immediate by induction on the derivation. □

Next, we define *legality* for expressions, environments and contexts, which in each case means that they are 'accepted' in some derivation.

Definition 3.5. (Legal expression, legal environment, legal combination, legal context)

1. An expression M is called *legal*, if there exist an environment Δ, a context Γ, and N such that $\Delta; \Gamma \vdash M : N$ or $\Delta; \Gamma \vdash N : M$. (For such Δ and Γ, we call M legal *with respect to* Δ *and* Γ.)

2. An environment Δ is called *legal*, if there exist a context Γ, and M and N such that $\Delta \,;\, \Gamma \vdash M : N$.

3. An environment Δ and a context Γ form a *legal combination*, if there exist M and N such that $\Delta \,;\, \Gamma \vdash M : N$.

4. A context Γ is called *legal*, if there exist an environment Δ, and M and N such that $\Delta \,;\, \Gamma \vdash M : N$.

We have the following properties of legality.

Lemma 3.6. *(Legality Lemma)*

1. *If $\Delta \equiv \Delta_1, \Delta_2$ and Δ is legal, then Δ_1 is legal.*

2. *If $\Gamma \equiv \Gamma_1, \Gamma_2$ and Γ is legal, then Γ_1 is legal.*

Proof. Each of the statements is immediate by induction on the derivation. □

Lemma 3.7. *(Start Lemma for declarations and definitions)*

1. *(Start for contexts) If* $\Delta;\Gamma$ *is a legal combination and* $(x : A) \in \Gamma$*, then we have* $\Delta\,;\,\Gamma \vdash x : A$*.*

2. *(Start for environments) Let* $\mathcal{D} \equiv \overline{x} : \overline{A} \rhd a(\overline{x}) := M : N$*. If* Δ *is legal and* $\mathcal{D} \in \Delta$*, then both (2a)* $\Delta\,;\,\overline{x} : \overline{A} \vdash M : N$ *and (2b)* $\Delta\,;\,\overline{x} : \overline{A} \vdash a(\overline{x}) : N$*.*

Proof. The proofs are immediate by induction on the derivation. □

There is more to be said about legal environments:

Lemma 3.8. *(Legal Environment Lemma)*
If $\Delta \equiv \Delta_1, \overline{x} : \overline{A} \rhd a(x) := M/\perp\!\!\!\perp : N, \Delta_2$ *and* $\Gamma = \overline{y} : \overline{B}$ *form a legal combination, then:*

1. *Each* A_i *is legal with respect to* Δ_1 *and* $x_1 : A_1, \ldots, x_{i-1} : A_{i-1}$ *and each* B_j *is legal with respect to* $\Delta; y_1 : B_1, \ldots, x_{j-1} : B_{j-1}$*.*

2. *Both* M *and* N *are legal with respect to* Δ_1 *and* $\overline{x} : \overline{A}$*.*

Proof. As Δ and Γ form a legal combination, we have $\Delta;\Gamma \vdash P : Q$ for some P, Q. We prove the statements simultaneously by induction on the derivation of this judgment. □

Legality is also used in the following lemma. It states that, if a judgment $M : N$ is derivable in an environment Δ and context Γ, then it remains derivable if we add more definitions to Δ or more declarations to Γ. Of course, this only holds if the extended Δ' and Γ' are still a legal combination. We also have to be precise of what we mean with an extension. We therefore extend the subset relation $\subseteq$ to environments and contexts.

Definition 3.9. For Δ and Δ' environments, we say that Δ *is a sub-environment of* Δ', notation $\Delta \subseteq \Delta'$ if $\overline{x} : \overline{A} \rhd a(\overline{x}) := M/\perp\!\!\!\perp : N \in \Delta$ implies $\overline{x} : \overline{A} \rhd a(\overline{x}) := M/\perp\!\!\!\perp : N \in \Delta'$ for all definitions.

For Γ and Γ' contexts, we say that Γ *is a sub-context of* Γ', notation $\Gamma \subseteq \Gamma'$ if $x : A \in \Gamma$ implies $x : A \in \Gamma'$ for all declarations.

Especially note that, if $\Gamma \subseteq \Gamma'$, then the declarations in Γ and Γ' need not be in the same order. Similarly for the definitions in Δ and Δ' in case $\Delta \subseteq \Delta'$.

Lemma 3.10. *(Thinning Lemma) Let* $\Delta \subseteq \Delta'$*,* $\Gamma \subseteq \Gamma'$*, and let* $\Delta';\Gamma'$ *be a legal combination. If* $\Delta\,;\,\Gamma \vdash M : N$*, then* $\Delta'\,;\,\Gamma' \vdash M : N$*.*

Proof. The proof is by induction on the derivation of $\Delta\,;\,\Gamma \vdash M : N$. The interesting cases are (*weak*) (and (*var*) is similar), (*form*), (*abst*) and (*def*) (and (*def-prim*) is similar). We treat these interesting cases.

- (*weak*) $\dfrac{\Delta\ ;\ \Gamma\ \vdash\ A : B \quad \Delta\ ;\ \Gamma\ \vdash\ C : s}{\Delta\ ;\ \Gamma,\ x : C\ \vdash\ A : B}$ if $x \notin \Gamma$
  Suppose $\Delta \subseteq \Delta'$ and $\Gamma,\ x : C \subseteq \Gamma'$ and $\Delta';\Gamma'$ is a legal combination. By IH (applied to the left premise) we immediately conclude $\Delta'\ ;\ \Gamma'\ \vdash\ A : B$ and we are done.

- (*form*) $\dfrac{\Delta\ ;\ \Gamma\ \vdash\ A : s_1 \quad \Delta\ ;\ \Gamma,\ x : A\ \vdash\ B : s_2}{\Delta\ ;\ \Gamma\ \vdash\ \Pi x : A\,.\,B\ :\ s_2}$
  Suppose $\Delta \subseteq \Delta'$ and $\Gamma, \subseteq \Gamma'$ and $\Delta'\ ;\ \Gamma'$ is a legal combination. By IH, $\Delta'\ ;\ \Gamma'\ \vdash\ A : s_1$, so $\Delta'\ ;\ \Gamma', x : A \vdash x : A$ by one application of the rule (*var*), so $\Delta'\ ;\ \Gamma', x : A$ is a legal combination. We apply the IH to the second premise and conclude $\Delta'\ ;\ \Gamma',\ x : A\ \vdash\ B : s_2$. Now, with one application of (*form*) we conclude $\Delta'\ ;\ \Gamma'\ \vdash\ \Pi x : A\,.\,B\ :\ s_2$.

- $(abst)\ \dfrac{\Delta\ ;\ \Gamma,\ x : A\ \vdash\ M : B \qquad \Delta\ ;\ \Gamma\ \vdash\ \Pi x : A\,.\,B\ :\ s}{\Delta\ ;\ \Gamma\ \vdash\ \lambda x : A\,.\,M\ :\ \Pi x : A\,.\,B}$
  Suppose $\Delta \subseteq \Delta'$ and $\Gamma, \subseteq \Gamma'$ and $\Delta'\ ;\ \Gamma'$ is a legal combination. By IH, $\Delta'\ ;\ \Gamma' \vdash\ \Pi x : A\,.\,B\ :\ s$. By Lemma 3.4, there is a sub-derivation of $\Delta\ ;\ \Gamma,\ \vdash\ A : s'$ (for some sort s'), so by IH $\Delta'\ ;\ \Gamma',\ \vdash\ A : s'$, so $\Delta'\ ;\ \Gamma', x : A\ \vdash\ x : A$, so Δ' and $\Gamma', x : A$ form a legal combination. Therefore we can apply IH to the left premise and we get $\Delta'\ ;\ \Gamma',\ x : A\ \vdash\ M : B$. With one application of $(abst)$ we conclude $\Delta'\ ;\ \Gamma'\ \vdash\ \lambda x : A\,.\,M\ :\ \Pi x : A\,.\,B$.

- $(def)\ \dfrac{\Delta\ ;\ \Gamma\ \vdash\ K : L \qquad \Delta\ ;\ \overline{x} : \overline{A}\ \vdash\ M : N}{\Delta\ ,\ \overline{x} : \overline{A} \rhd a(\overline{x}) := M : N\ ;\ \Gamma\ \vdash\ K : L}$ if $a \notin \Delta$
  Suppose $\Delta\ ,\ \overline{x} : \overline{A} \rhd a(\overline{x}) := M : N\ \subseteq \Delta'$ and $\Gamma, \subseteq \Gamma'$ and $\Delta'\ ;\ \Gamma'$ is a legal combination. By IH, $\Delta'\ ;\ \Gamma'\ \vdash\ K : L$, so we are done.

□

We first prove an important *Generation Lemma*, that breaks down a derivable judgment in its premises.

Lemma 3.11. *(Generation Lemma)*

1. *If $\Delta;\Gamma \vdash x : C$, then there is a sort s and an expression B, such that $B \overset{\Delta}{=}_\beta C$ and $\Delta;\Gamma \vdash B : s$ and $x : B \in \Gamma$.*

2. *If $\Delta;\Gamma \vdash MN : C$, then there are A and B such that $\Delta;\Gamma \vdash M : \Pi x : A\,.\,B$, $\Delta;\Gamma \vdash N : A$, and $C \overset{\Delta}{=}_\beta B[x := N]$.*

3. *If $\Delta;\Gamma \vdash \lambda x : A\,.\,b : C$, then there are a sort s and an expression B such that $C \overset{\Delta}{=}_\beta \Pi x : A\,.\,B$ and $\Delta;\Gamma \vdash \Pi x : A\,.\,B : s$, and $\Delta;\Gamma, x : A \vdash b : B$.*

4. *If $\Delta;\Gamma \vdash \Pi x : A\,.\,B : C$, then there are sorts s_1 and s_2 such that $C \overset{\Delta}{=}_\beta s_2$, and $\Delta;\Gamma \vdash A : s_1$ and $\Delta;\Gamma, x : A \vdash B : s_2$.*

5. *If $\Delta;\Gamma \vdash a(\overline{U}) : C$, then $\Delta \equiv \Delta_1, \mathcal{D}, \Delta_2$ with $\mathcal{D} \equiv \overline{x} : \overline{A} \rhd a(\overline{x}) := M/\perp\!\!\!\perp : N$ for some M, N and $C \overset{\Delta}{=}_\beta N[\overline{x} := \overline{U}]$. Moreover, if $|\overline{A}| > 0$, then $\Delta;\Gamma \vdash \overline{U} : \overline{A[\overline{x} := \overline{U}]}$ and:*
   - *if $\mathcal{D}$ is a descriptive definition, then $\Delta;\Gamma \vdash M : N$;*
   - *if $\mathcal{D}$ is a primitive definition, then $\Delta;\Gamma \vdash N : s$ for some sort s.*

Proof. The proof is by induction on the derivation. As a matter of fact, the lemma just inverts the deduction rules and applies weakening (or thinning): it considers the shape of the subject term and describes how it could have been derived. This is by the rule for forming a term of that shape, followed by applications of conversion and weakening, that don't change the subject, but only the context, the environment or the type.

1. If $\Delta;\Gamma \vdash x : C$, the last applied rule is (var), $(conv)$, (def),$(def\text{-}prim)$ or $(weak)$. The first is immediate; the second is immediate by induction on the derivation and (def),$(def\text{-}prim)$ are immediate by IH and one application of the rule.

   For $(weak)$: $\dfrac{\Delta\ ;\ \Gamma'\ \vdash\ x : C \quad \Delta\ ;\ \Gamma'\ \vdash\ A : s'}{\Delta\ ;\ \Gamma',\ y : A\ \vdash\ x : C}$ if $y \notin \Gamma'$

By IH there is a sort s and an expression B, such that $B \stackrel{\Delta}{=}_\beta C$ and $\Delta;\Gamma' \vdash B : s$ and $x : B \in \Gamma'$. So by (*weak*): $\Delta;\Gamma',\ y : A \vdash B : s$ and we are done.

2. If $\Delta;\Gamma \vdash MN : C$, the last applied rule is (*appl*), (*conv*), (*def*),(*def-prim*) or (*weak*). The first is immediate; the second is immediate by induction on the derivation and (*def*),(*def-prim*) are immediate by IH and one application of the rule.

   For (*weak*): $\dfrac{\Delta\ ;\ \Gamma' \ \vdash\ MN : C \quad \Delta\ ;\ \Gamma' \ \vdash\ A : s'}{\Delta\ ;\ \Gamma',\ y : A \ \vdash\ MN : C}$ if $y \notin \Gamma'$

   By IH there are A and B such that $\Delta;\Gamma' \vdash M : \Pi x : A\,.\,B$, $\Delta;\Gamma' \vdash N : A$, and $C \stackrel{\Delta}{=}_\beta B[x := N]$. Now, by twice weakening, we conclude $\Delta;\Gamma',\ y : A \vdash M : \Pi x : A\,.\,B$, $\Delta;\Gamma',\ y : A \vdash N : A$, and we are done

3. If $\Delta;\Gamma \vdash \lambda x : A\,.\,b : C$, the last applied rule is (*abst*), (*conv*), (*def*),(*def-prim*) or (*weak*). The first is immediate; the second is immediate by induction on the derivation and (*def*),(*def-prim*) are immediate by IH and one application of the rule.

   For (*weak*): $\dfrac{\Delta\ ;\ \Gamma' \ \vdash\ \lambda x : A\,.\,b : C \quad \Delta\ ;\ \Gamma' \ \vdash\ D : s'}{\Delta\ ;\ \Gamma',\ y : D \ \vdash\ \lambda x : A\,.\,b : C}$

   By IH there are a sort s and a B such that $C \stackrel{\Delta}{=}_\beta \Pi x : A\,.\,B$ and $\Delta;\Gamma' \vdash \Pi x : A\,.\,B : s$, and $\Delta;\Gamma', x : A \vdash b : B$. By (*weak*) we conclude that also $\Delta;\Gamma',\ y : D \vdash \Pi x : A\,.\,B : s$. By the Context Lemma 3.4 we derive that $\Delta;\Gamma' \vdash A : s''$ for some s'', so, by (*weak*), also $\Delta;\Gamma',\ y : D \vdash A : s''$, so $\Delta;\Gamma',\ y : D,\ x : A \vdash x : A$. So, Δ and $\Gamma',\ y : D,\ x : A$ are a legal combination, and thus we can apply the Thinning Lemma 3.10 to derive $\Delta;\Gamma', y : D, x : A \vdash b : B$ (from $\Delta;\Gamma', x : A \vdash b : B$), and we are done.

4. The proof for $\Delta;\Gamma \vdash \Pi x : A\,.\,B : C$ is totally similar (but a bit simpler) to case (3).

5. If $\Delta;\Gamma \vdash a(\overline{U}) : C$, the last applied rule is (*inst*), (*inst-prim*), (*conv*), (*def*),(*def-prim*) or (*weak*). The first two are immediate; the third is immediate by induction on the derivation and (*def*),(*def-prim*) are immediate by IH and one application of the rule.

   For (*weak*) we do the proof for a descriptive definition, as the one for a primitive definition is similar:

   $$\frac{\Delta\ ;\ \Gamma' \ \vdash\ a(\overline{U}) : C \quad \Delta\ ;\ \Gamma' \ \vdash\ D : s'}{\Delta\ ;\ \Gamma',\ y : D \ \vdash\ a(\overline{U}) : C}$$

   By IH: $\Delta \equiv \Delta_1, \mathcal{D}, \Delta_2$ with $\mathcal{D} \equiv \overline{x} : \overline{A} \rhd a(\overline{x}) := M/\perp\!\!\!\perp : N$ for some M, N and $C \stackrel{\Delta}{=}_\beta N[\overline{x} := \overline{U}]$. Moreover $\Delta;\Gamma \vdash M : N$ and if $|\overline{A}| > 0$, then $\Delta;\Gamma \vdash \overline{U} : \overline{A[\overline{x} := \overline{U}]}$.

   By one application of (*weak*) we conclude that $\Delta;\Gamma,\ y : D \vdash M : N$ and, if $|\overline{A}| > 0$, then $\Delta;\Gamma,\ y : D \vdash \overline{U} : \overline{A[\overline{x} := \overline{U}]}$ and we are done.

□

Lemma 3.12. *(Substitution Lemma) For Δ and $\Gamma_1, x : A, \Gamma_2$ a legal combination, P, B and N terms,*

$$\left.\begin{array}{r}\Delta;\Gamma_1, x : A, \Gamma_2 \vdash P : B \\ \Delta;\Gamma_1 \vdash N : A\end{array}\right\} \implies \Delta;\Gamma_1, \Gamma_2[x := N] \vdash P[x := N] : B[x := N].$$

Proof. By induction on the derivation of $\Delta;\Gamma_1, x:A, \Gamma_2 \vdash P : B$, assuming that $\Delta;\Gamma_1 \vdash N : A$ is derivable. The only cases that are somewhat interesting is when the last rule is (*appl*), (*inst*) or (*inst-prim*), because some attention has to be given to the substitutions and renaming of variables. We treat the (*appl*). the other cases are similar, but require more variable administration.

$$(appl)\ \frac{\Delta\ ;\ \Gamma_1, x:A, \Gamma_2\ \vdash\ M\ :\ \Pi y:B\,.\,C \qquad \Delta\ ;\ \Gamma_1, x:A, \Gamma_2\ \vdash\ P : B}{\Delta\ ;\ \Gamma_1, x:A, \Gamma_2\ \vdash\ MN\ :\ C[y:=N]}$$

Now by IH and (*appl*):
$\Delta;\Gamma_1, \Gamma_2[x:=N] \vdash M[x:=N]P[x:=N] : C[y:=N][y:=P[x:=N]]$.
We may assume that $y \notin FV(\Gamma_1, x:A, \Gamma_2)$ (because it is a bound variable that we otherwise rename.) Hence $y \notin FV(N)$ and so we can conclude $C[y:=N][y:=P[x:=N]] \equiv (C[y:=P])[y:=N]$ and we are done. □

Lemma 3.13. *(Correctness of Types Lemma) For Δ and Γ a legal combination, M and A terms,*

$$\Delta;\Gamma \vdash P : A \implies \exists s \in \{*, \Box\}[A \equiv s \vee \Delta;\Gamma \vdash A : s].$$

Proof. The proof is by induction on the derivation of $\Delta;\Gamma \vdash P : A$. The only cases that are interesting are when the last rule is (*appl*), (*inst*) or (*inst-prim*).

- $(appl)\ \dfrac{\Delta\ ;\ \Gamma\ \vdash\ M\ :\ \Pi x:A\,.\,B \qquad \Delta\ ;\ \Gamma\ \vdash\ N : A}{\Delta\ ;\ \Gamma\ \vdash\ MN\ :\ B[x:=N]}$

  Then $\Delta;\Gamma \vdash \Pi x:A.B : s$ by IH and hence by Generation Lemma, $\Delta;\Gamma, x:A \vdash B : s$. Now by the Substitution Lemma, we conclude that $\Delta;\Gamma \vdash B[x:=N] : s$.

- $(inst)\ \dfrac{\Delta\,;\,\Gamma \vdash * : \Box \quad \Delta\,;\,\Gamma \vdash \overline{U} : \overline{A[\overline{x}:=\overline{U}]}}{\Delta\ ;\ \Gamma\ \vdash\ a(\overline{U}) : N[\overline{x}:=\overline{U}]}$

  with $\overline{x} : \overline{A} \rhd a(\overline{x}) := M : N \in \Delta$.
  Then there is a sub-derivation of $\Delta;\overline{x} : \overline{A} \vdash M : N$. We apply IH to conclude that $N \equiv s$ (and we are done) or $\Delta;\overline{x} : \overline{A} \vdash N : s$ for some s. By the Substitution Lemma: $\Delta;\overline{x} : \overline{A} \vdash [\overline{x} := \overline{U}] : s$

- $(inst\text{-}prim)\ \dfrac{\Delta\,;\,\Gamma \vdash * : \Box \quad \Delta\,;\,\Gamma \vdash \overline{U} : \overline{A[\overline{x}:=\overline{U}]}}{\Delta\ ;\ \Gamma\ \vdash\ a(\overline{U}) : N[\overline{x}:=\overline{U}]}$

  with $\overline{x} : \overline{A} \rhd a(\overline{x}) := \perp\!\!\!\perp : N \in \Delta$.
  Then there is a sub-derivation of $\Delta;\overline{x} : \overline{A} \vdash N : s$ for some s. By the Substitution Lemma: $\Delta;\overline{x} : \overline{A} \vdash [\overline{x} := \overline{U}] : s$

□

Lemma 3.14. *(Uniqueness of Types Lemma) If Δ, Γ is a legal combination, M, C and C' are terms we have*

$$\left.\begin{array}{l}\Gamma \vdash M : C\\ \Gamma \vdash M : C'\end{array}\right\} \implies C =^{\Delta}_{\beta} C'.$$

Proof. By induction on the structure of the term M, using the Generation Lemma. The only interesting case is when M is an application, so we do that case: let $M \equiv PN$. Then we find

by Generation Lemma A, A', B and B' such that

$$\begin{aligned}\Delta;\Gamma &\vdash P : \Pi x : A.B,\\ \Delta;\Gamma &\vdash P : \Pi x : A'.B',\end{aligned}$$

with $C \stackrel{\Delta}{=}_\beta B[x := N]$, $C' \stackrel{\Delta}{=}_\beta B'[x := N]$. We apply IH to conclude that $\Pi x : A.B \stackrel{\Delta}{=}_\beta \Pi x : A'.B'$. By the confluence property for $\stackrel{\Delta}{\twoheadrightarrow}_\beta$, to be proved in the next Section (Theorem 4.7), we conclude from this that $B \stackrel{\Delta}{=}_\beta B'$ and hence $B[x := N] \stackrel{\Delta}{=}_\beta B'[x := N]$ and we are done. □

We now prove the important Subject Reduction Lemma, which states that, if $P : D$ and P reduces to P', then also $P' : D$. Typed terms 'do not go wrong', as Milner would say. To prove this, we need to extend this to contexts, because of the rules (*abst*) and (*form*).

Lemma 3.15. *(Subject Reduction Lemma) For Δ an environment and Γ and Γ' contexts, P, P' and D terms,*

$$\begin{aligned}\Delta;\Gamma \vdash P : D \wedge P \stackrel{\Delta}{\twoheadrightarrow}_\beta P' &\implies \Delta;\Gamma \vdash P' : D,\\ \Delta;\Gamma \vdash P : D \wedge \Gamma \stackrel{\Delta}{\twoheadrightarrow}_\beta \Gamma' &\implies \Delta;\Gamma' \vdash P : D.\end{aligned}$$

Proof. The proof of the two statements is done simultaneously, by induction on the derivation of $\Delta;\Gamma \vdash P : D$, distinguishing cases according to the last rule. We only prove the first statement.

All cases except for the last rule being (*appl*) are immediate. (For (*form*) and (*abst*), use IH on the second statement.)

If the last rule is (*appl*), we distinguish sub-cases according to where the reduction takes place.

Subcase 1

$$\frac{\Delta;\Gamma \vdash M : \Pi x : A.C \quad \Delta;\Gamma \vdash N : A}{\Delta;\Gamma \vdash MN : C[x := N]}$$

with $P \equiv MN$ and the reduction is inside M or N. Then we are immediately done by IH.

Subcase 2

$$\frac{\Delta;\Gamma \vdash \lambda x : A.M : \Pi x : B.C \quad \Delta;\Gamma \vdash N : B}{\Delta;\Gamma \vdash (\lambda x : A.M)N : C[x := N]}$$

with $P \equiv (\lambda x : A.M)N$ and $P' \equiv M[x := N]$. Then by the Generation Lemma applied to the first premise, we find

$$\begin{aligned}\Delta;\Gamma, x : A \vdash M : C' \quad (1)\\ \Delta;\Gamma \vdash \Pi x : A.C' : s\\ \Pi x : A.C' \stackrel{\Delta}{=}_\beta \Pi x : B.C\end{aligned}$$

So, again by Generation

$$\begin{aligned}\Delta;\Gamma \vdash A : s_1 (\text{for some } s_1) \quad (2)\\ \Delta;\Gamma, x : A \vdash C' : s\end{aligned}$$

By the Church-Rosser property, we conclude from $\Pi x : A.C' \stackrel{\Delta}{=}_\beta \Pi x : B.C$ that

$$A \stackrel{\Delta}{=}_\beta B \quad (3)$$
$$C' \stackrel{\Delta}{=}_\beta C. \quad (4)$$

So, applying (*conv*) to (2) and the left premise, using (3), we get

$$\Delta; \Gamma \vdash N : A. \quad (5)$$

Applying the Substitution Lemma to (5) and (1) we get

$$\Delta; \Gamma \vdash M[x := N] : C'[x := N]. \quad (6)$$

By applying the Correctness of Types Lemma 3.13 to the first premise, we find $\Delta; \Gamma \vdash \Pi x : B.C : s$, hence by Generation

$$\Delta; \Gamma, x : B \vdash C : s. \quad (7)$$

Now apply Substitution to (3) and (7) to get

$$\Delta; \Gamma \vdash C[x := N] : s. \quad (8)$$

Apply (*conv*) to (6) and (8) (using (4)) to conclude

$$\Delta; \Gamma \vdash M[x := N] : C[x := N].$$

□

Corollary 3.16. *(Type Reduction Lemma) If $\Delta; \Gamma \vdash P : B$ and $B \stackrel{\Delta}{\twoheadrightarrow}_\beta C$, then $\Delta; \Gamma \vdash P : C$.*

Proof. By the Correctness of Types Lemma 3.13, $\Delta; \Gamma \vdash B : s$ or $B \equiv s$,for some s. In the first case, we are done by one application of (*conv*). In the second case we are done immediately. □

The opposite of Thinning is *Condensing*, also called *Strengthening*: if a definition or a declaration is not used in a term, then we can remove it and still have a derivable judgment. The proof of this property is remarkably tricky, although the property seems to be obviously correct. We first need a sub-lemma. The idea of using this Sub-lemma to prove Condensing is due to [6], who used it for the system ECC. The author and Nederhof used it in paper [5] to give the proof of Condensing (there called Strengthening) for so called *functional PTSs*.

Lemma 3.17. *(Sub-lemma to prove Condensing)*

1. *If Δ is an environment, $\Gamma_1, x : A, \Gamma_2$ a context and M and B are terms, then*

$$\left.\begin{array}{r} \Delta; \Gamma_1, x : A, \Gamma_2 \vdash M : B \\ x \notin FV(\Gamma_2, M) \end{array}\right\} \Longrightarrow \exists B'[B \stackrel{\Delta}{\twoheadrightarrow}_\beta B' \wedge \Delta; \Gamma_1, \Gamma_2 \vdash M : B'].$$

2. *If $\Delta_1, a(\overline{x}) := M/\!\!\perp\!\!\perp : N, \Delta_2$ is an environment, Γ a context and M and B are terms, then*

$$\left.\begin{array}{r} \Delta_1, a(\overline{x}) := M/\!\!\perp\!\!\perp : N, \Delta_2; \Gamma \vdash M : B \\ a \notin (\Delta_2, \Gamma, M) \end{array}\right\} \Longrightarrow \exists B'[B \stackrel{\Delta}{\twoheadrightarrow}_\beta B' \wedge \Delta_1, \Delta_2; \Gamma \vdash M : B'].$$

Proof. The proof of both statements is by induction on the derivation of $\Delta; \Gamma_1, x : A, \Gamma_2 \vdash M : B$, respectively $\Delta_1, a(\overline{x}) := M/\perp\!\!\!\perp : N, \Delta_2; \Gamma \vdash M : B$. We distinguish cases according to the last rule. The arguments for (1) and (2) are the same and the only interesting cases are when the last rule is (*abst*), (*appl*), (*conv*) or (*inst*), so we treat those.

(*abst*) Say $M \equiv \lambda y : C.N$, $B \equiv \Pi y : C.D$ and

$$\frac{\Delta; \Gamma_1, x : A, \Gamma_2, y : C \vdash N : D \quad \Delta; \Gamma_1, x : A, \Gamma_2 \vdash \Pi y : C.D : s}{\Delta; \Gamma_1, x : A, \Gamma_2 \vdash \lambda y : C.N : \Pi y : C.D}$$

Then by IH $\Delta; \Gamma_1, \Gamma_2, y : C \vdash N : D'$ for some D' with $D \overset{\Delta}{\twoheadrightarrow}_\beta D'$.
Also, for some s', $\Delta; \Gamma_1, x : A, \Gamma_2 \vdash C : s'$ is a conclusion of a sub-derivation of the derivation with conclusion $\Delta; \Gamma_1, x : A, \Gamma_2, y : C \vdash N : D$, so by IH $\Delta; \Gamma_1, \Gamma_2 \vdash C : s'$.
By Correctness of Types we find that, for some s'', $\Delta; \Gamma_1, \Gamma_2, y : C \vdash D' : s''$ or $D' \equiv s'$.
In the second case too there is an s_2 such that $\Delta; \Gamma_1, \Gamma_2, y : C \vdash D' (\equiv s'') : s_2$, because for D there is such s_2 and we have SR and Uniqueness of Types. So, in any case $\Delta; \Gamma_1, \Gamma_2, y : C \vdash D' : s''$ for some s''.
So we can apply (*form*) to conclude $\Delta; \Gamma_1, \Gamma_2 \vdash \Pi y : C.D' : s$ and hence $\Delta; \Gamma_1, \Gamma_2 \vdash \lambda y : C.N : \Pi y : C.D'$.

(*appl*) Say $M \equiv NP$, $B \equiv D[y := P]$ and

$$\frac{\Delta; \Gamma_1, x : A, \Gamma_2 \vdash N : \Pi y : C.D \quad \Delta; \Gamma_1, x : A, \Gamma_2 \vdash P : C}{\Delta; \Gamma_1, x : A, \Gamma_2 \vdash NP : D[y := P]}$$

Then by IH, $\Delta; \Gamma_1, \Gamma_2 \vdash N : \Pi y : C'.D'$ and $\Delta; \Gamma_1, \Gamma_2 \vdash N : C''$ with $C \overset{\Delta}{\twoheadrightarrow}_\beta C', C''$ and $D \overset{\Delta}{\twoheadrightarrow}_\beta D'$. By Church-Rosser we find a term C''' such that $C', C'' \overset{\Delta}{\twoheadrightarrow}_\beta C'''$ and hence (by Corollary 3.16) $\Delta; \Gamma_1, \Gamma_2 \vdash N : \Pi y : C'''.D'$ and $\Delta; \Gamma_1, \Gamma_2 \vdash P : C'''$. We may now conclude that $\Delta; \Gamma_1, \Gamma_2 \vdash NP : D'[y := P]$ and we are done.

(*conv*) Say

$$\frac{\Delta; \Gamma_1, x : A, \Gamma_2 \vdash M : C \quad \Delta; \Gamma_1, x : A, \Gamma_2 \vdash D : s}{\Delta; \Gamma_1, x : A, \Gamma_2 \vdash M : D} \; C \overset{\Delta}{=}_\beta D$$

Then by IH $\Delta; \Gamma_1, \Gamma_2 \vdash M : C'$ for some C' with $C \overset{\Delta}{\twoheadrightarrow}_\beta C'$. By confluence (Theorem 4.7), there is a C'' such that $D \overset{\Delta}{\twoheadrightarrow}_\beta C''$ and $D \overset{\Delta}{\twoheadrightarrow}_\beta C''$. Now $\Delta; \Gamma_1, \Gamma_2 \vdash M : C''$ and we are done.

(*inst*) Say

$$\frac{\Delta \,;\, \Gamma_1, x : A, \Gamma_2 \vdash * : \Box \quad \Delta \,;\, \Gamma_1, x : A, \Gamma_2 \vdash \overline{U} : \overline{B[\overline{x} := \overline{U}]}}{\Delta \;;\; \Gamma_1, x : A, \Gamma_2 \;\vdash\; a(\overline{U}) : N[\overline{x} := \overline{U}]}$$

with $\overline{x} : \overline{B} \rhd a(\overline{x}) := M : N \in \Delta$.
By IH, $\Delta \,;\, \Gamma_1, \Gamma_2 \vdash \overline{U} : \overline{C}$ for some $\overline{C}$ with $\overline{B[\overline{x} := \overline{U}]} \overset{\Delta}{\twoheadrightarrow}_\beta \overline{C}$. As $x \notin \mathrm{FV}(\overline{B[\overline{x} := \overline{U}]})$, we can conclude that $\overline{B[\overline{x} := \overline{U}]}$ is well-typed and apply (*conv*) to conclude $\Delta \,;\, \Gamma_1, \Gamma_2 \vdash \overline{U} : \overline{B[\overline{x} := \overline{U}]}$. So also $\Delta \;;\; \Gamma_1, \Gamma_2 \;\vdash\; a(\overline{U}) : N[\overline{x} := \overline{U}]$

□

Lemma 3.18. *(Condensing Lemma)*

1. *If Δ is an environment, $\Gamma_1, x : A, \Gamma_2$ a context and M and B are terms, then*

$$\left.\begin{array}{r} \Delta; \Gamma_1, x : A, \Gamma_2 \vdash M : B \\ x \notin FV(\Gamma_2, M, B) \end{array}\right\} \Longrightarrow \Delta; \Gamma_1, \Gamma_2 \vdash M : B.$$

2. *If $\Delta_1, a(\overline{x}) := M/\perp\!\!\!\perp : N, \Delta_2$ is an environment, Γ a context and M and B are terms, then*

$$\left.\begin{array}{r} \Delta_1, a(\overline{x}) := M/\perp\!\!\!\perp : N, \Delta_2; \Gamma \vdash M : B \\ a \notin (\Delta_2, \Gamma, M, B) \end{array}\right\} \Longrightarrow \Delta_1, \Delta_2; \Gamma \vdash M : B.$$

Proof. We prove (1), because (2) is similar. By the Sub-lemma we find a B' such that $B \overset{\Delta}{\twoheadrightarrow}_\beta B'$ and

$$\Delta; \Gamma_1, \Gamma_2 \vdash M : B'.$$

By Correctness of Types there are two possibilities, $\Delta; \Gamma_1, x : A, \Gamma_2 \vdash B : s$ or $B \equiv s$ for some s. In the second case we are immediately done, because $B \equiv B'$. In the first case we can once again apply the Sub-lemma to

$$\Delta; \Gamma_1, x :, A, \Gamma_2 \vdash B : s$$

to find that

$$\Delta; \Gamma_1, \Gamma_2 \vdash B : s.$$

Now we are done by one application of (*conv*). □

4 Confluence and Normalization in λD

We now prove confluence and normalization properties of λD. Normalization is another word for termination and comes in two 'flavors': we recall that a term M is *weakly normalizing* if there *exists* a convertible term N which is in normal form; M is *strongly normalizing* if all reduction paths starting from M are *finite*.

First we consider the new relation $\overset{\Delta}{\rightarrow}$, which formalizes the unfolding of a definition. Given a legal expression M in λD, does there always exist a δ-reduction path starting with M which terminates? And if $M \overset{\Delta}{\twoheadrightarrow} M_1$ and $M \overset{\Delta}{\twoheadrightarrow} M_2$, is there an M_3 such that $M_1 \overset{\Delta}{\twoheadrightarrow} M_3$ and $M_2 \overset{\Delta}{\twoheadrightarrow} M_3$? This is indeed the case. We give a prove by defining, for a term M, the term $|M|_\Delta$, and showing that $M \overset{\Delta}{\twoheadrightarrow} |M|_\Delta$, that $|M|_\Delta$ is a Δ-normal form and that, if $M \overset{\Delta}{\twoheadrightarrow} M_1$, then $|M|_\Delta \equiv |M|_\Delta$. This method of proof is strongly related to the proof of a similar result in [8], where it is applied to a different definition calculus. (With local definitions, but no parametrized definitions.)

For the confluence and the normalization of $\overset{\Delta}{\rightarrow}$, we don't need well-typedness: we prove these properties for expressions $\mathcal{E}_{\lambda D}$. We do need some restrictions on the definitions (e.g. to avoid circular definitions) for these proofs, but we don't need well-typedness. (Well-typedness will only be used for strong normalization of $\beta\delta$, Section 4.2.) We therefore introduce the notion of *non-circular environment*.

Definition 4.1. (Non-circular environment) An environment $\Delta = a_1(\overline{x}) := M_1 : N_1, \ldots, a_n(\overline{x}) := M_n : N_n$ is *non-circular* if for all $1 \leq j \leq n$, a_j does not occur in $\{M_1, N_1, \ldots, M_{j-1}, N_{j-1}\}$.

Definition 4.2. Given a non-circular environment Δ and an expression P that only uses constants from Δ, we define $|P|_\Delta$ by induction on the sum of the lengths of Δ and P.

$$\begin{array}{rcl} |x|_\Delta & := & x \\ |PQ|_\Delta & := & |P|_\Delta |Q|_\Delta \\ |\lambda x : A.P|_\Delta & := & \lambda x : |A|_\Delta . |P|_\Delta \\ |\Pi x : A.B|_\Delta & := & \Pi x : |A|_\Delta . |B|_\Delta \\ |a(\overline{U})|_\Delta & := & |M|_{\Delta_1}[\overline{x} := |\overline{U}|_\Delta] \\ & & \text{if } \Delta = \Delta_1, \overline{x} : \overline{A} \rhd a(\overline{x}) := M : N, \Delta_2 \end{array}$$

The idea of $|P|_\Delta$ is that it unfolds all definitions that occur in P. An easy fact is that, if Δ and Δ' are non-circular environments and $\Delta \subseteq \Delta'$, then $|M|_\Delta = |M|_{\Delta'}$.

We have the following properties of $|-|_\Delta$. First we state a basic Substitution Lemma for $|-|_\Delta$, that will be used in the more important Lemma that states the crucial properties of $|-|_\Delta$.

Lemma 4.3. *(Substitution Lemma for $|-|_\Delta$.) For P a term, $\overline{x}$ a sequence of variables and $\overline{U}$ a sequence of terms of the same length as $\overline{x}$:*

$$|P[\overline{x} := \overline{U}]|_\Delta \equiv |P|_\Delta[\overline{x} := |\overline{U}|_\Delta].$$

Proof. By induction on the sum of the lengths of Δ and P. □

Lemma 4.4. *Let Δ be a non-circular environment and let P and N be terms such that $\Delta; \Gamma \vdash P : B$ for some Γ and B.*

1. *If $P \overset{\Delta}{\to} N$, then $|P|_\Delta \equiv |N|_\Delta$.*
2. *If $P \to_\beta N$, then $|P|_\Delta \twoheadrightarrow_\beta |N|_\Delta$.*
3. *$P \overset{\Delta}{\twoheadrightarrow} |P|_\Delta$.*
4. *$|P|_\Delta$ is in δ-normal form.*

Proof. 1. The proof is by induction on the sum of the lengths of Δ and P. The only interesting case is when $P = a(\overline{U})$, with a a descriptive definition and $N = M[\overline{x} := \overline{U}]$. Then $|P|_\Delta = |M|_{\Delta_1}[\overline{x} := |\overline{U}|_\Delta]$ and we need a substitution property: $|M|_{\Delta_1}[\overline{x} := |\overline{U}|_\Delta] \equiv |M[\overline{x} := \overline{U}]|_\Delta$, but this is just Lemma 4.3.

2. By induction on the sum of the lengths of Δ and P. In each case, a one step $\to_\beta$ in P produces a one step $\to_\beta$ in $|P|_\Delta$, except for the case where $Pa(\overline{U})$, with a a descriptive definition and the $\to_\beta$-step is in $\overline{U}$. Depending on the definition of a, this $\to_\beta$-step may get copies or may be thrown away.

3. By induction on the sum of the lengths of Δ and P.

4. By induction on the sum of the lengths of Δ and P, one shows that $|P|_\Delta$ cannot contain a definite description.

□

Theorem 4.5. *(Weak Normalization of $\overset{\Delta}{\to}$) The reduction $\overset{\Delta}{\to}$ is weakly normalizing for λD. That is: given an expression P in $\mathcal{E}_{\lambda D}$ that only use constants from Δ, there is an N in $\overset{\Delta}{\to}$-normal form such that $P \overset{\Delta}{\twoheadrightarrow} N$.*

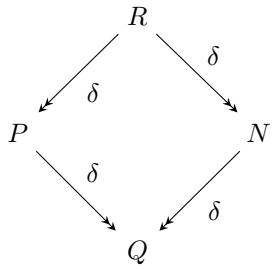
Proof. Given P an expressions that only uses constants from Δ, take $N := |P|_\Delta$. Then we are done by Lemma 4.4, parts (3) and (4). □

Theorem 4.6. *(Church–Rosser (CR) for $\overset{\Delta}{\to}$; confluence for δ)*

1. *The reduction $\overset{\Delta}{\to}$ is Church–Rosser for λD.*
   That is: given an expression P in $\mathcal{E}_{\lambda D}$ that only use constants from Δ, if $P \overset{\Delta}{\twoheadrightarrow} P_1$ and $P \overset{\Delta}{\twoheadrightarrow} P_2$, then there is an P_3 such that $P_1 \overset{\Delta}{\twoheadrightarrow} P_3$ and $P_2 \overset{\Delta}{\twoheadrightarrow} P_3$.

2. *The reduction $\overset{\Delta}{\to}$ is confluent for λD.*
   That is: given expressions P and N in $\mathcal{E}_{\lambda D}$ that only use constants from Δ, if $P \overset{\Delta}{=} N$, then there is a Q such that $P \overset{\Delta}{\twoheadrightarrow} Q$ and $N \overset{\Delta}{\twoheadrightarrow} Q$.

Proof. For (1), Take $P_3 : |P|_\Delta$, then we are done by part (1) of 4.4. For (2), we may assume that the conversion path from P to N only passes through expressions that only use constants from Δ. So, the standard proof to derive confluence from CR also applies here, as illustrated by the following diagram.



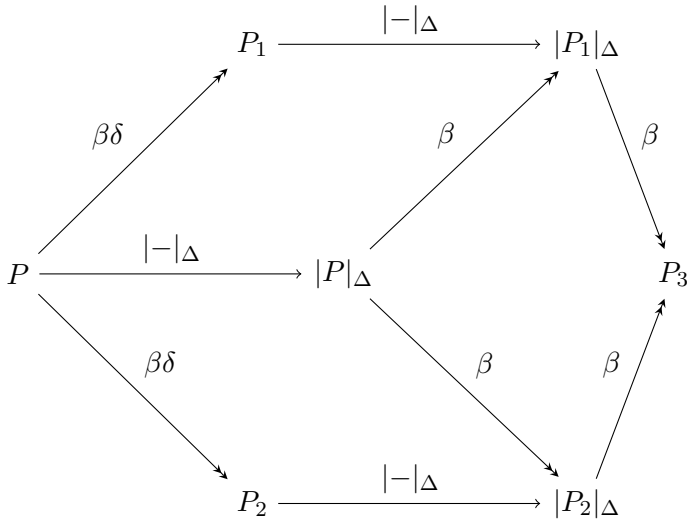
□

Theorem 4.7. *(Church–Rosser (CR) for $\overset{\Delta}{\twoheadrightarrow_\beta}$; $\beta\delta$-confluence)*

1. *The reduction $\overset{\Delta}{\twoheadrightarrow_\beta}$ is Church–Rosser.*
   That is: given an expression P in $\mathcal{E}_{\lambda D}$ that only use constants from Δ, if $P \overset{\Delta}{\twoheadrightarrow_\beta} P_1$ and $P \overset{\Delta}{\twoheadrightarrow_\beta} P_2$, then there is an P_3 such that $P_1 \overset{\Delta}{\twoheadrightarrow_\beta} P_3$ and $P_2 \overset{\Delta}{\twoheadrightarrow_\beta} P_3$.

2. *The reduction $\overset{\Delta}{\twoheadrightarrow_\beta}$ is confluent.*
   That is: given expressions P and N in $\mathcal{E}_{\lambda D}$ that only use constants from Δ, if $P \overset{\Delta}{=_\beta} N$, then there is a Q such that $P \overset{\Delta}{\twoheadrightarrow_\beta} Q$ and $N \overset{\Delta}{\twoheadrightarrow_\beta} Q$.

Proof. Part (1) is by the following diagram chase.



If $P \overset{\Delta}{\twoheadrightarrow}_\beta P_1$, then $|P|_\Delta \twoheadrightarrow_\beta |P_1|_\Delta$, by parts (1) and (2) of Lemma 4.4. Similarly, $|P|_\Delta \twoheadrightarrow_\beta |P_2|_\Delta$, so the first two tilted rectangles commute. The existence of a P_3 such that $|P_1|_\Delta \twoheadrightarrow_\beta P_3$ and $|P_2|_\Delta \twoheadrightarrow_\beta P_3$ follows from confluence of β-reduction on $\mathcal{E}_{\lambda D}$. By (3) of Lemma 4.4 we conclude that $P_1 \overset{\Delta}{\twoheadrightarrow}_\beta P_3$ and $P_2 \overset{\Delta}{\twoheadrightarrow}_\beta P_3$.

Part (2) follows from part (1) in the same way as for Theorem 4.6. □

A consequence of $\beta\delta$-confluence is uniqueness of normal forms:

Corollary 4.8. *(Uniqueness of $\beta\delta$-normal form) If P is a well-typed term in λD that has a $\beta\delta$-normal form, then this normal form is unique.*

A final Corollary that we can draw from $|-|_\Delta$ and its properties is that $\overset{\Delta}{\twoheadrightarrow}_\beta$ is weakly normalizing. We also need the fact that the Calculus of Constructions is weakly normalizing. It is also strongly normalizing, which we will be using later, see [4].

Theorem 4.9. *(Weak Normalization for $\overset{\Delta}{\twoheadrightarrow}_\beta$ in λD) If P is legal in λD, then there is a (finite) $\beta\delta$-reduction sequence starting from P that ends in a $\beta\delta$-normal form.*

Proof. Suppose $\Delta;\Gamma \vdash P : B$. Then $P \overset{\Delta}{\twoheadrightarrow} |P|_\Delta$, which is in δ-normal form. So $|P|_\Delta$ does not contain descriptive definitions. It may contain primitive definitions $a(\overline{x}) := \perp\!\!\!\perp$, but these pose no problem for (strong) normalization. (Cf. e.g. the proof of [4].) So, by normalization of the Calculus of Constructions, we conclude that $|P|_\Delta$ has a β-normal form. This term is still in δ-normal form, so we have obtained a $\beta\delta$-normal form of P. □

4.1 Strong normalization for $\overset{\Delta}{\to}$

It requires more effort to prove that strong normalization holds for $\overset{\Delta}{\to}$. We use the method of Recursive Path Ordering (RPO) (see [2, 9]) to prove strong normalization of $\overset{\Delta}{\to}$.

Definition 4.10. (Multiset RPO ordering) Let $\mathcal{F}$ be a set of function symbols, each with fixed arity, $\mathcal{X}$ be a countable set of variables and let $\mathrm{term}(\mathcal{F},\mathcal{X})$ be the set of (open) terms over $\mathcal{F}$ and $\mathcal{X}$ (respecting the arities). Let $>_\mathcal{F}$ be a partial order on $\mathcal{F}$.

The *multiset recursive path order* $>_{\mathrm{rpo}}$ *on* $\mathrm{term}(\mathcal{F},\mathcal{X})$ *induced by* $>_\mathcal{F}$ is defined by

1. for each j, $f(s_1, \ldots, s_m) >_{\text{rpo}} s_j$
2. for each j, if $s_j >_{\text{rpo}} t$, then $f(s_1, \ldots, s_m) >_{\text{rpo}} t$,
3. if $f >_{\mathcal{F}} g$ and $\forall j[f(s_1, \ldots, s_m) >_{\text{rpo}} t_j]$, then $f(s_1, \ldots, s_m) >_{\text{rpo}} g(t_1, \ldots, t_n)$,
4. if $\forall j[f(s_1, \ldots, s_m) >_{\text{rpo}} t_j]$ and $\langle s_1, \ldots, s_m\rangle >_{\text{rpo}}^{\text{mul}} \langle t_1, \ldots, t_m\rangle$ then $f(s_1, \ldots, s_m) >_{\text{rpo}} f(t_1, \ldots, t_m)$.

Here, $>_{\text{rpo}}^{\text{mul}}$ denotes the *multiset extension of* $>_{\text{rpo}}$, that is $\langle s_1, \ldots, s_m\rangle >_{\text{rpo}}^{\text{mul}} \langle t_1, \ldots, t_m\rangle$ if the *multiset* $\{\{t_1, \ldots, t_m\}\}$ can be obtained from $\{\{s_1, \ldots, s_m\}\}$ by finitely many times replacing an s_j with finitely many t such that $s_j >_{\text{rpo}} t$.

An easy consequence of this definition is the following: If $s_i >_{\text{rpo}} t_i$, then $f(s_1, \ldots, s_m) >_{\text{rpo}} f(s_1, \ldots, s_{i-1}, t_i, s_{i+1}, \ldots, s_m)$. This follows from rules (4) and (1) in the definition of $>_{\text{rpo}}$.

Theorem 4.11. *([2]) Let $>_{\mathcal{F}}$ be a partial order on the set of function symbols $\mathcal{F}$ and let $>_{\text{rpo}}$ be the induced multiset recursive path order. Then*

$$>_{\text{rpo}} \text{ is well-founded} \iff >_{\mathcal{F}} \text{ is well-founded.}$$

Proof. See [2] or [9]. □

Given a non-circular environment Δ, we choose a set of function symbols $\mathcal{F}$ and a relation $>_{\mathcal{F}}$ that is well-founded on $\mathcal{F}$ and we define a map $\mathcal{T}$ from terms of λD only involve constants from Δ to term such that $M \overset{\Delta}{\to} N$ implies $\mathcal{T}(M) >_{\text{rpo}} \mathcal{T}(N)$. Then we conclude that $\overset{\Delta}{\to}$ is strongly normalizing.

Remark 4.12. *In the rest of this subsection, let $\Delta = a_1(\overline{x}) := M_1 : N_1, \ldots a_n(\overline{x}) := M_n : N_n$ be a fixed non-circular environment. We are proving normalization, so we may assume that Δ only contains descriptive definitions, as primitive definitions do not reduce.*

In the rest of this section, we will be concerned with expressions in $\mathcal{E}_{\lambda D}$ that only use constants from Δ. So we will not be relying on well-typedness, but only on the well-foundedness of the definitions (non-circularity).

Definition 4.13. We define the set of functions symbols $\mathcal{F}$ as $\{a_1, \ldots, a_n, \lambda, \Pi, \text{app}\}$, where each a_i has the arity it is given in the definition in Δ (i.e. the length of $\overline{x}$), and the arities of λ, Π and app are 2. The partial ordering $>_{\mathcal{F}}$ on $\mathcal{F}$ is defined by

$$a_n >_{\mathcal{F}} a_{n_1} >_{\mathcal{F}} \ldots >_{\mathcal{F}} a_1 >_{\mathcal{F}} \lambda, \Pi, \text{app}.$$

Definition 4.14. We define the mapping $\mathcal{T}$ from $\mathcal{E}_{\lambda D}$ term$(\mathcal{F}, V)$ by

$$\begin{aligned}
\mathcal{T}(x) &:= x \\
\mathcal{T}(\lambda x : A.P) &:= \lambda(\mathcal{T}(A), \mathcal{T}(P)) \\
\mathcal{T}(\Pi x : A.P) &:= \Pi(\mathcal{T}(A), \mathcal{T}(P)) \\
\mathcal{T}(PQ) &:= \text{app}(\mathcal{T}(P), \mathcal{T}(Q)) \\
\mathcal{T}(a(\overline{U}) &:= a(\overline{\mathcal{T}(U)})
\end{aligned}$$

Lemma 4.15. *For $a_i(\overline{U})$ an expression in $\mathcal{E}_{\lambda D}$ that only uses constants from Δ, $\mathcal{T}(a_i(\overline{U})) >_{\text{rpo}} \mathcal{T}(M_i[\overline{x} := \overline{U}])$.*

Proof. The proof is by induction on M_i.

If M_i is a λ-abstraction, a Π-abstraction, an application or $a_k(\overline{V})$ for some a_k and $\overline{V}$, then $a_i >_{\mathcal{F}} f$ for f the top function symbol of $\mathcal{T}(M_i[\overline{x} := \overline{U}])$. (For the a_k case, this follows from the fact that M_i can only contain constants $a_1, \ldots, a_{i-1}$.) So to prove $\mathcal{T}(_i(\overline{U})) >_{\text{rpo}} \mathcal{T}(M_i[\overline{x} := \overline{U}])$, we only have to prove $\mathcal{T}(_i(\overline{U})) >_{\text{rpo}} \mathcal{T}(t[\overline{x} := \overline{U}])$ for t a sub-term of M_i, which holds by induction hypothesis.

If $M_i = x$, a variable, then $\mathcal{T}(a_i(\overline{U})) >_{\text{rpo}} \mathcal{T}(U_i)$ immediately. □

Lemma 4.16. *For P an expression in $\mathcal{E}_{\lambda D}$ that only uses constants from Δ, if $P \xrightarrow{\Delta} Q$, then $\mathcal{T}(P) >_{\text{rpo}} \mathcal{T}(Q)$.*

Proof. By induction on the structure of P. All cases follow immediately from the induction hypothesis, except when $P = a_i(\overline{U}) \xrightarrow{\Delta} M_i[\overline{x} := \overline{U}] = Q$. This case follows from Lemma 4.15 □

Theorem 4.17. *(Strong Normalization of $\xrightarrow{\Delta}$) For Δ a non-circular environment, the relation $\xrightarrow{\Delta}$ is strongly normalizing.*

Proof. Given Δ, we define $\mathcal{F}$ and $>_{\mathcal{F}}$ as in Definition 4.13 Then $>_{\text{rpo}}$ is well-founded, due to Theorem 4.11. But then $\xrightarrow{\Delta}$ is well-founded, because an infinite $\xrightarrow{\Delta}$-reduction path would give an infinite descending $>_{\text{rpo}}$-path by Lemma 4.16. □

4.2 Strong Normalization for $\xrightarrow{\Delta}_{\beta}$

We now prove strong normalization for the combination of β and δ. We do this by defining a reduction preserving translation to a type system (without definitions) λC^{∞}, that is known to be strongly normalizing. The proof does not rely on the fact that $\xrightarrow{\Delta}$ is strongly normalizing; as a matter of fact, strong normalization of $\xrightarrow{\Delta}$ is a consequence of the result in the present section. (So, in a sense the previous section is superfluous, but we felt that a more 'primitive' proof of strong normalization of $\xrightarrow{\Delta}$ would be in place.)

The proof in this section is an adaptation of the proof in [8]. We use the terminology of *Pure Type Systems*, see [1, 3], to define our notions.

Definition 4.18. We define λC^{∞} as the Pure Type System with

$$\begin{aligned} \mathcal{S} &:= \{*, \Box_0, \Box_1, \ldots\}, \\ \mathcal{A} &:= \{* : \Box_0, \Box_0 : \Box_1 : \ldots\}, \\ \mathcal{R} &:= \{(s, *, *) \mid s \in \mathcal{S}\} \cup \{(\Box_i, \Box_j, \Box_{\max(i,j)}) \mid i, j \geq 0\}. \end{aligned}$$

The important things of λC^{∞} are:

1. A Π-type can always be formed and thus, every λ-abstraction is legal: if $\Gamma, x : A \vdash M : B$, then $\Gamma \vdash \lambda x : A.M : \Pi x : A.B$.
2. Every type is well-typed: if $\Gamma, x : A \vdash M : B$, then $\Gamma, x : A \vdash B : s$ for some $s \in \mathcal{S}$.

The crucial property of λC^∞ is that it is strongly normalizing.

Theorem 4.19. *([6]) In the system λC^∞, β-reduction is strongly normalizing.*

Proof. In [6], the system ECC is introduced and proven to be strongly normalizing. The system λC^∞ is a subsystem of ECC, and thus also strongly normalizing. □

Definition 4.20. We define the map $\{-\}_\Delta$ from terms in λD to terms in λC^∞ as follows. (The map straightforwardly extends to sequences of terms.)

$$\begin{array}{rcl}
\{x\}_\Delta & := & x \\
\{PQ\}_\Delta & := & \{P\}_\Delta\{Q\}_\Delta \\
\{\lambda x:A.P\}_\Delta & := & \lambda x:\{A\}_\Delta.\{P\}_\Delta \\
\{\Pi x:A.B\}_\Delta & := & \Pi x:\{A\}_\Delta.\{B\}_\Delta \\
\{a(\overline{U})\}_\Delta & := & (\lambda\overline{x}:\overline{A}.\{M\}_\Delta)\{\overline{U}\}_\Delta \\
 & & \text{if } \Delta = \Delta_1, \overline{x}:\overline{A} \rhd a(\overline{x}) := M : N, \Delta_2 \\
\{a(\,)\}_\Delta & := & (\lambda x_0 : T_0.\{M\}_\Delta)t_0, \\
 & & \text{if } \Delta = \Delta_1, \rhd a(\,) := M : N, \Delta_2.
\end{array}$$

Here, t_0 and T_0 are some arbitrary closed terms in λC^∞ such that $\vdash t_0 : T_0$, and x_0 is an arbitrary variable that does not occur anywhere else. (The $a(\,)$ case denotes a descriptive definition without arguments.)

Lemma 4.21. *For $\Delta;\Gamma \vdash P : B$, if $P \stackrel{\Delta}{\to}_\beta Q$, then $\{P\}_\Delta \twoheadrightarrow^+_\beta \{Q\}_\Delta$ (which denotes a non-empty reduction sequence).*

Proof. The proof in immediate by induction on the structure of P. If $P \to_\beta Q$, then $\{P\}_\Delta \to_\beta \{Q\}_\Delta$. If $P \stackrel{\Delta}{\to} Q$, then possibly $\{P\}_\Delta \twoheadrightarrow^+_\beta \{Q\}_\Delta$. □

Theorem 4.22. *(Strong Normalisation for $\stackrel{\Delta}{\to}_\beta$ in λD) For each legal Δ, the relation $\stackrel{\Delta}{\to}_\beta$ is strongly normalizing.*

Proof. Given P with $\Delta;\Gamma P : B$, if there is an infinite $\beta\delta$-reduction sequence starting from P, then, due to Lemma 4.21 there is an infinite β-reduction sequence starting from $\{P\}_\Delta$ in λC^∞. But this is impossible, so we are done. □

References

[1] H. Barendregt. Lambda calculi with types. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*. Oxford Univ. Press, 1993.

[2] N. Dershowitz. A note on simplification orderings. *Inf. Process. Lett.*, 9(5):212–215, 1979.

[3] H. Geuvers. Logics and type systems, PhD thesis, Radboud University Nijmegen, the Netherlands, 1993.

[4] H. Geuvers. A short and flexible proof of strong normalization for the calculus of constructions. In Peter Dybjer, Bengt Nordström, and Jan M. Smith, editors, *Types for Proofs and Programs, International Workshop TYPES'94, Båstad, Sweden, June 6-10, 1994, Selected Papers*, volume 996 of *Lecture Notes in Computer Science*, pages 14–38. Springer, 1995.

[5] H. Geuvers and M.-J. Nederhof. Modular proof of strong normalization for the calculus of constructions. *J. Funct. Program.*, 1(2):155–189, 1991.

[6] Z. Luo. Ecc, an extended calculus of constructions. In *Proceedings, Fourth Annual Symposium on Logic in Computer Science, 5-8 June, 1989, Asilomar Conference Center, Pacific Grove, California, USA*, pages 386–395, 1989.

[7] R. Nederpelt and H. Geuvers. *Type Theory and Formal Proof, An Introduction*. Cambridge University Press, 2014.

[8] P. Severi and E. Poll. Pure type systems with definitions. In Anil Nerode and Yuri Matiyasevich, editors, *Logical Foundations of Computer Science, Third International Symposium, LFCS'94, St. Petersburg, Russia, July 11-14, 1994, Proceedings*, volume 813 of *Lecture Notes in Computer Science*, pages 316–328. Springer, 1994.

[9] H. Zantema. Termination of term rewriting by semantic labelling. *Fundam. Inform.*, 24(1/2):89–105, 1995.